

NAME : HARSHIT SINGHANIA
ROLL : 2105890
LAB ASSIGNMENT 3

LAs

1 WAP using C to Evaluate the Given Polynomial Equation $f(x)$. Note: Order of polynomial, co-efficient and value of x will be user input.

```
#include <stdio.h>
#include <conio.h>
float poly(float a[], int, float);
int main()
{
    float x, a[10], y1;
    int deg, i;
    printf("Enter the degree of polynomial equation: ");
    scanf("%d", &deg);
    printf("Enter the value of x for which the equation is to be evaluated: ");
    scanf("%f", &x);
    for (i = 0; i <= deg; i++)
    {
        printf("Enter the coefficient of x to the power %d: ", i);
        scanf("%f", &a[i]);
    }
    y1 = poly(a, deg, x);
    printf("The value of polynomial equation for the value of x = %.2f is: %.2f", x, y1);
    return 0;
}
/* function for finding the value of polynomial at some value of x */
float poly(float a[], int deg, float x)
{
    float p;
    int i;
    p = a[deg];
    for (i = deg; i >= 1; i--)
    {
        p = (a[i - 1] + x * p);
    }
    return p;
}
```

```

Enter the degree of polynomial equation: 2
Enter the value of x for which the equation is to be evaluated: 1
Enter the coefficient of x to the power 0: 1
Enter the coefficient of x to the power 1: 1
Enter the coefficient of x to the power 2: 1
The value of polynomial equation for the value of x = 1.00 is: 3.00

```

2. WAP using function that adds given two polynomials $f(x) = h(x) + g(x)$

```
#include <stdio.h>
```

```

struct poly
{
    int coeff;
    int expo;
};
struct poly p1[10], p2[10], p3[10];
int readPoly(struct poly[]);
int addPoly(struct poly[], struct poly[], int, int, struct poly[]);
void displayPoly(struct poly[], int terms);

```

```

int main()
{
    int t1, t2, t3;

    t1 = readPoly(p1);
    printf(" \n First polynomial : ");
    displayPoly(p1, t1);
    t2 = readPoly(p2);
    printf(" \n Second polynomial : ");
    displayPoly(p2, t2);
    t3 = addPoly(p1, p2, t1, t2, p3);
    printf(" \n\n Resultant polynomial after addition : ");
    displayPoly(p3, t3);
    printf("\n");

    return 0;
}

```

```

int readPoly(struct poly p[10])
{
    int t1, i;

    printf("\n\n Enter the total number of terms in the polynomial:");
    scanf("%d", &t1);

```

```

printf("\n Enter the COEFFICIENT and EXPONENT in DESCENDING ORDER\n");
for (i = 0; i < t1; i++)
{
    printf("  Enter the Coefficient(%d): ", i + 1);
    scanf("%d", &p[i].coeff);
    printf("    Enter the exponent(%d): ", i + 1);
    scanf("%d", &p[i].expo);
}
return (t1);
}

```

```

int addPoly(struct poly p1[10], struct poly p2[10], int t1, int t2, struct poly p3[10])
{
    int i, j, k;

    i = 0;
    j = 0;
    k = 0;

    while (i < t1 && j < t2)
    {
        if (p1[i].expo == p2[j].expo)
        {
            p3[k].coeff = p1[i].coeff + p2[j].coeff;
            p3[k].expo = p1[i].expo;

            i++;
            j++;
            k++;
        }
        else if (p1[i].expo > p2[j].expo)
        {
            p3[k].coeff = p1[i].coeff;
            p3[k].expo = p1[i].expo;
            i++;
            k++;
        }
        else
        {
            p3[k].coeff = p2[j].coeff;
            p3[k].expo = p2[j].expo;
            j++;
            k++;
        }
    }
}

```

```

    }
}

while (i < t1)
{
    p3[k].coeff = p1[i].coeff;
    p3[k].expo = p1[i].expo;
    i++;
    k++;
}
while (j < t2)
{
    p3[k].coeff = p2[j].coeff;
    p3[k].expo = p2[j].expo;
    j++;
    k++;
}

return (k);
}

void displayPoly(struct poly p[10], int term)
{
    int k;

    for (k = 0; k < term - 1; k++)
        printf("%d(x^%d)+", p[k].coeff, p[k].expo);
    printf("%d(x^%d)", p[term - 1].coeff, p[term - 1].expo);
}

```

```

Enter the COEFFICIENT and EXPONENT in DESCENDING ORDER
Enter the Coefficient(1): 32
Enter the exponent(1): 12
Enter the Coefficient(2): 1
Enter the exponent(2): 2
Enter the Coefficient(3): 34
Enter the exponent(3): 3

First polynomial : 32(x^12)+1(x^2)+34(x^3)

Enter the total number of terms in the polynomial:3

Enter the COEFFICIENT and EXPONENT in DESCENDING ORDER
Enter the Coefficient(1): 22
Enter the exponent(1): 12
Enter the Coefficient(2): 3
Enter the exponent(2): 2
Enter the Coefficient(3): 34
Enter the exponent(3): 3

Second polynomial : 22(x^12)+3(x^2)+34(x^3)

Resultant polynomial after addition : 54(x^12)+4(x^2)+68(x^3)
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22>

```

3. WAP to check whether the given matrix is sparse matrix or not.

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int row, col, i, j, a[10][10], count = 0;
    printf("Enter row\n");
    scanf("%d", &row);
    printf("Enter Column\n");
    scanf("%d", &col);
    printf("Enter Element of Matrix1\n");
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    printf("Elements are:\n");
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)

```

```

    {
        printf("%d\t", a[i][j]);
    }
    printf("\n");
}
for (i = 0; i < row; i++)
{
    for (j = 0; j < col; j++)
    {
        if (a[i][j] == 0)
            count++;
    }
}
if (count > ((row * col) / 2))
    printf("Matrix is a sparse matrix \n");
else
    printf("Matrix is not sparse matrix\n");
}

```

```

PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22> cd "c:\Users\KIIT\Desktop\1-8-22\" ; if ($?) { gcc 3.c -o 3 } ; if ($?) { .\3 }
Enter row
3
Enter Column
3
Enter Element of Matrix1
1
0
0
0
0
0
0
0
0
0
Elements are:
1      0      0
0      0      0
0      0      0
Matrix is a sparse matrix
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22> █

```

4. WAP to find the transpose of a matrix.

```

#include <stdio.h>
int main()

```

```

{
    int a[10][10], transpose[10][10], r, c;
    printf("Enter rows and columns: ");
    scanf("%d %d", &r, &c);

    printf("\nEnter matrix elements:\n");
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j)
        {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }

    printf("\nEnter matrix: \n");
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j)
        {
            printf("%d ", a[i][j]);
            if (j == c - 1)
                printf("\n");
        }

    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j)
        {
            transpose[j][i] = a[i][j];
        }

    printf("\nTranspose of the matrix:\n");
    for (int i = 0; i < c; ++i)
        for (int j = 0; j < r; ++j)
        {
            printf("%d ", transpose[i][j]);
            if (j == r - 1)
                printf("\n");
        }
    return 0;
}

```

```
Enter rows and columns: 3 3
```

```
Enter matrix elements:
```

```
Enter element a11: 1
```

```
Enter element a12: 2
```

```
Enter element a13: 3
```

```
Enter element a21: 4
```

```
Enter element a22: 5
```

```
Enter element a23: 6
```

```
Enter element a31: 7
```

```
Enter element a32: 8
```

```
Enter element a33: 9
```

```
Entered matrix:
```

```
1  2  3
```

```
4  5  6
```

```
7  8  9
```

```
Transpose of the matrix:
```

```
1  4  7
```

```
2  5  8
```

```
3  6  9
```

```
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22>
```

5. WAP to find determinant of 3x3 Matrix.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int a[3][3], i, j;
```

```
    long determinant;
```

```
    printf("Enter the 9 elements of matrix: ");
```

```
    for (i = 0; i < 3; i++)
```

```
        for (j = 0; j < 3; j++)
```

```
            scanf("%d", &a[i][j]);
```

```
    printf("\nThe matrix is\n");
```

```
    for (i = 0; i < 3; i++)
```

```
    {
```

```
        printf("\n");
```

```
        for (j = 0; j < 3; j++)
```

```
            printf("%d\t", a[i][j]);
```



```

    }

    determinant = a[0][0] * ((a[1][1] * a[2][2]) - (a[2][1] * a[1][2])) - a[0][1] * (a[1][0] * a[2][2] - a[2][0]
    * a[1][2]) + a[0][2] * (a[1][0] * a[2][1] - a[2][0] * a[1][1]);

    printf("\nDeterminant of 3X3 matrix: %ld", determinant);

    return 0;
}

```

```
Enter the 9 elements of matrix: 1 2 3 4 5 6 7 8 9
```

```
The matrix is
```

```

1      2      3
4      5      6
7      8      9

```

```
Determinant of 3X3 matrix: 0
```

```
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22> █
```

6. WAP to Find Largest Element in an Array using Recursion.

```
#include <stdio.h>
```

```
#define N 5
```

```

void biggest(int *num, int n, int big)
{
    if (n < 0)
        printf("Biggest element is %d\n", big);
    else
    {
        if (*num > big)
            big = *num;

        biggest(++num, --n, big);
    }
}

```

```

int main()
{
    int a[N], i;

    printf("Enter %d integer numbers\n", N);

```

```

    for (i = 0; i < N; i++)
        scanf("%d", &a[i]);

    biggest(a, N - 1, a[0]);

    return 0;
}

```

```

PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22> cd "c:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22\" ; if ($?) { gcc 6.c -o 6 } ; if ($?) { .\6 }
Enter 5 integer numbers
1
2
4
3
5
Biggest element is 5
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22>

```

7. WAP using function to find frequency of occurrence of numbers in an array.

```
#include <stdio.h>
```

```

int search(int a[],int n,int key)
{
    int count = 0, i;
    for (int i = 0; i < n; i++)
        if (a[i] == key)
            count++;
    return count;
}

int main()
{
    int arr[100];
    int n;
    int find;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
    {
        printf("enter %d element:", i + 1);
        scanf("%d", &arr[i]);
    }
    printf("enter key:");
    scanf("%d", &find);
}

```

```

int call = search(arr, n, find);
printf("%d", call);
return 0;
}

```

```

Enter number of elements: 10
enter 1 element:1
enter 2 element:1
enter 3 element:2
enter 4 element:2
enter 5 element:3
enter 6 element:3
enter 7 element:4
enter 8 element:4
enter 9 element:4
enter 10 element:5
enter key:4
3
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22>

```

8. WAP to determine whether the given matrix is a lower triangular or upper triangular or tri-diagonal matrix.

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int row, col, i, j;
    printf("enter row:");
    scanf("%d", &row);
    printf("enter column:");
    scanf("%d", &col);
    int a[row][col];
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("\nEnter a[%d][%d] value :: ", i, j);
            scanf("%d", &a[i][j]);
        }
    }
    printf("\n Check wether matrix is upper triangular or not \n");
}

```

```

int flag1 = 1;
for (i = 0; i < row; i++)
{
    for (j = 0; j < i; j++)
    {
        if (a[i][j] != 0)
        {
            flag1 = 0;
        }
    }
}
flag1 = 1;

```

```

if (flag1 == 0)
    printf("it is not upper triangular");
else if (flag1 == 1)
    printf("it is upper triangular");

```

```

printf("\n Check wether matrix is lower triangular or not \n");
int flag2 = 1;
for (i = 0; i < row - 1; i++)
    for (j = i + 1; j < col; j++)
        if (a[i][j] != 0)
            flag2 = 0;
flag2 = 1;

```

```

if (flag2 == 0)
    printf("it is not lower triangular");
else if (flag2 == 1)
    printf("it is lower triangular");

```

```

printf("\n Check wether matrix is tri diagonal or not \n");
int flag3 = 1;
for (i=0; i<row; i++)
    for (j=0; j<col; j++)
        if (i==j)
            if (a[i][j]==0)
                flag3=0;
            else if (a[i][j+1]==0)
                flag3=0;
            else if (a[i+1][j]==0)
                flag3=0;
flag3=1;

```

```

    if (flag3 == 0)
        printf("it is not tri diagonal");
    else if (flag3 == 1)
        printf("it is tri diagonal");
}

```

```

PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22> cd "c:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22"
{ gcc 8iii.c -o 8iii } ; if ($?) { .\8iii }
enter row:3
enter column:3

Enter a[0][0] value :: 1
Enter a[0][1] value :: 2
Enter a[0][2] value :: 3
Enter a[1][0] value :: 4
Enter a[1][1] value :: 5
Enter a[1][2] value :: 6
Enter a[2][0] value :: 7
Enter a[2][1] value :: 8
Enter a[2][2] value :: 9

Check wether matrix is upper triangular or not
it is upper triangular
Check wether matrix is lower triangular or not
it is lower triangular
Check wether matrix is tri diagonal or not
it is tri diagonal
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\1-8-22>

```